What Is Claimed Is:

1. A method of processing loop instructions using a data processing device having a central processing unit (CPU) and a coprocessor, wherein the CPU fetches and decodes instructions retrieved from program memory and determines whether the instructions are CPU-type or coprocessor-type, comprising the steps of:

decoding the coprocessor-type instructions by the coprocessor and if a loop operation is decoded, retrieving from the program memory the instructions within the loop;

storing the retrieved instructions within the loop in a loop buffer; and

inhibiting instruction fetch from the program memory while instructions within the loop are executed in a subsequent iteration of the loop.

2. The method of Claim 1, further including the step of accessing the instructions within the loop from the loop buffer in a subsequent iteration of the loop.

3. The method of Claim 1, wherein said step of decoding further includes determining a backward branch distance for use by the CPU to control branching to and from the loop.

4. The method of Claim 1 further including the steps of:

determining from the loop instruction a number of iterations of the loop operation;

decrementing by the coprocessor the number of iterations upon completion of each loop; and

signaling to the CPU the completion of the loop operation when reaching the end of the number of iterations.

5. The method of Claim 1, wherein said storing step includes storing 'n' loop instructions in 'm' registers of the loop buffer and addressing the 'm' registers by $\log_2 m$ least significant bits (LSBs) of a program counter which is also used for addressing the program memory, wherein n or m is any natural number and n is less than or equal to m.

6. The method of Claim 5, further including the steps of accessing the instructions stored in the loop buffer through a multiplexer and controlling the multiplexer output by the $\log_2 m$ LSBs of the program counter.

7. The method of Claim 5, wherein a first instruction within the loop is stored in any of the m registers addressed by the LSBs of the program counter.

8. The method of Claim 1, further including the steps of signaling the presence or absence of an active loop instruction by a loop buffer flag in each of the 'm' registers in the loop buffer, the presence of an active instruction in a register is indicated by a preassigned signal in the loop buffer flag.

9. The method of Claim 8, further including the step of accessing each flag in the loop buffer by $\log_2 m$ least significant bits of a program counter used for addressing the program memory.

10. The method of Claim 8, further including the step of multiplexing an instruction from the loop buffer and the program memory, the multiplexing is dependent upon a presence of an active instruction signal from a loop buffer flag.

11. The method of Claim 8, wherein said step of inhibiting instruction fetch from the program memory includes sending an inhibit signal to the program memory when the preassigned signal in the loop buffer flag is read and indicates the presence of an active loop instruction.

12. The method of Claim 11, wherein the preassigned signal in each of said loop buffers is selectively alterable by the CPU independent of the presence or absence of an active instruction in corresponding registers.

13. The method of Claim 8, further including the step of clearing the loop buffer flag when the loop operation is completed.

14. A data processing device comprising:

a central processing unit (CPU) for fetching instructions from a program memory, decoding the instructions and sending a signal (CCLK) to a coprocessor if a coprocessor type instruction is decoded;

a coprocessor for decoding the coprocessor-type instructions upon receipt of the signal (CCLK); and

a loop buffer for receiving from the program memory instructions within a loop and storing the instructions within the loop when the coprocessor decodes a loop

operation from the coprocessor-type instructions, wherein the instructions within the loop are retrieved from the loop buffer for execution in a subsequent iteration of the loop.

15. The device of Claim 14, wherein a disable signal is sent to the program memory for inhibiting access of the program memory while the instructions within the loop are retrieved from the loop buffer.

16. The device of Claim 14, wherein the loop buffer includes 'm' registers, each having a corresponding loop buffer flag for indicating whether the corresponding register is filled with an instruction.

17. The device of Claim 16, wherein the loop buffer flags are accessed by $\log_2 m$ least significant bits of a program counter used for addressing the program memory.

18. The device of Claim 16, wherein a program memory inhibit signal is generated based on a signal read from the loop buffer flag.

19. The device of Claim 14, wherein the loop buffer includes 'm' registers and the registers are addressed by $\log_2 m$ LSBs of a program counter used for addressing the program memory.

20. The device of Claim 14, further including a multiplexer for multiplexing between the instructions retrieved from the program memory and the instructions

retrieved from the loop buffer, the multiplexor being controlled by signals read from the loop buffer flags.

21. The device of Claim 14, wherein the coprocessor decodes from a loop instruction a loop block size and a number of iterations of looping, and calculates a backward branch distance for use by the CPU to control branching to and from the loop.

22. The device of Claim 21, wherein the backward branch distance is the loop block size minus one.

23. The device of Claim 14, wherein the instructions stored in the loop buffer comprise coprocessor and CPU type instructions.

24. A data processing device comprising:

a central processing unit (CPU) for fetching instructions from a program memory, decoding the instructions and sending a signal (CCLK) to a coprocessor if a coprocessor type instruction is decoded;

a coprocessor for decoding the coprocessor-type instructions upon receipt of the signal (CCLK); and

a loop buffer for receiving from the program memory instructions within a loop and storing the instructions within the loop when the coprocessor decodes a loop operation from the coprocessor-type instructions, wherein the instructions within the loop

are retrieve from the loop buffer for execution in a subsequent iteration of the loop, wherein a disable signal is sent to the program memory for inhibiting access of the program memory while the instructions within the loop are retrieved from the loop buffer.